14 日付

Created by GT F
Last updated 2019-08-17

クラス`Date`は、特定のインスタントを表すもので、その精度はミリ秒です。日付文字列のフォーマットと構文解析には`DateFormat`クラスをそれぞれ使用する必要があります。

DateFormat

`DateFormat`は、言語に依存しない方法で日付または時間をフォーマットおよび解析する、日時フォーマット・サブクラスの抽象クラスです。`SimpleDateFormat`などの日時フォーマット・サブクラスによって、フォーマット(日付→テキスト)、解析(テキスト→日付)および正規化を行うことができます。日付は、`Date`オブジェクトまたは1970年1月1日グリニッジ標準時00:00:00からのミリ秒で表現されます。

直接にDateオブジェクトをPrintすると、以下結果が出力されます。

```
1  // Sat Aug 17 11:17:32 GMT+09:00 2019
2  System.out.println(new Date());
```

`DateFormat`を用い、日付を指定するパターンで出力します。

```
1  SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd");
2  String dateString = simpleDateFormat.format(new Date());
3  System.out.println(dateString);
```

代表的なデータフォーマットは以下です。

パターン	例
yyyy-MM-dd	2019-01-01
yyyy-MM-dd HH:mm:ss	2019-01-01 11:00
yyyy/MM/dd	2019/01/01
yyyy/MM/dd HH:mm:ss	2019/01/01 11:22

なお、指定された文字列から`Date`を変換する可能です。

```
try {
    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd");
    Date date = simpleDateFormat.parse("2010-01-01");
    System.out.println(date);
} catch (ParseException ex) {
    Logger.getLogger(FileStreamSample.class.getName()).log(Level.SEVERE, null, ex);
}
```

※データフォーマットが不正の場合、ParseExceptionを発生します。

日付の操作

指定された２つ日付について、比較操作は可能です。

操作	説明
boolean before(Date another)	この日付が、指定された日付より前にあるかどうかを判定します
boolean after(Date another)	この日付が、指定された日付より後にあるかどうかを判定します

Calendar

Dateクラスはちょっと古いですが、現在のJava日付と時間フィールドの間の変換には`Calendar`クラスを、日付文字列のフォーマットと構文解析には`DateFormat`クラスをそれぞれ使用する必要があります。

```
Date currentDate = new Date();
System.out.println(currentDate);

// 日付 ⇒ カレンダーオブジェクト
Calendar c = Calendar.getInstance();
c.setTime(currentDate);

// 日付加算
c.add(Calendar.YEAR, 1);
c.add(Calendar.MONTH, 1);
c.add(Calendar.DATE, 1);
c.add(Calendar.HOUR, 1);
c.add(Calendar.MINUTE, 1);
c.add(Calendar.SECOND, 1);

// カレンダーオブジェクト ⇒ 日付
Date currentDatePlusOne = c.getTime();
System.out.println(currentDatePlusOne);
```

質問集

質問１：以下共通メソッドを作成してください。

1. メソッド名：eachEveryDay
2. 引数１：`yyyy-MM-dd` 開始日（文字列）
3. 引数２：`yyyy-MM-dd` 終了日（文字列）
4. 処理内容：開始日から終了日まで、ループして結果を`yyyy-MM-dd`で出力する。

質問２：勤務表の集計。以下CSVファイルは「**2019年06月度**」の勤務期間です。集計してください。

1. 毎日の作業時間（X時間Y分）。（15分切り）
2. 今月の総作業時間（X時間Y分）
3. 19:00以後は残業時間となり、毎日＆今月の総残業時間（X時間Y分）を集計してください。

勤務表のCSVファイルは以下通りです。

```
日付, 開始時刻, 終了時刻, 休憩時間
1, , ,
2, , ,
3, 10:00, 20:15, 1:00
4, 10:00, 20:00, 1:00
5, 10:00, 20:15, 1:00
6, 10:00, 20:00, 1:00
7, 10:00, 19:30, 1:00
8, , ,
9, , ,
10, 10:00, 21:30, 1:00
11, 10:00, 21:00, 1:00
12, 10:00, 21:30, 1:00
13, 10:00, 20:00, 1:00
14, , ,
15, , ,
16, , ,
17, 10:00, 20:00, 1:00
18, 10:00, 20:00, 1:00
19, , ,
20, 10:00, 20:30, 1:00
21, 9:00, 20:00, 1:00
22, , ,
23, , ,
24, 10:00, 21:30, 1:00
25, 10:00, 20:30, 1:00
26, 10:00, 21:00, 1:00
27, 10:00, 20:30, 1:00
28, 10:00, 19:15, 1:00
29, , ,
30, , ,
```

Like Be the first to like this

No labels